

An Optimized Computational Model for Multi-Community-Cloud Social Collaboration

Fei Hao^{1,2}, Geyong Min², Jinjun Chen³, Fei Wang¹, Man Lin⁴, Changqing Luo¹, and Laurence T. Yang^{1,4,*}



Abstract—Community Cloud Computing is an emerging and promising computing model for a specific community with common concerns, such as security, compliance and jurisdiction. It utilizes the spare resources of networked computers to provide the facilities so that the community gains services from the cloud. The effective collaboration among the community clouds offers a powerful computing capacity for complex tasks containing the subtasks that need data exchange. Selecting the best group of community clouds that are the most economy-efficient, communication-efficient, secured, and trusted to accomplish a complex task is very challenging. To address this problem, we firstly formulate a computational model for multi-community-cloud collaboration, namely *MC*³. The proposed model is then optimized from four aspects: minimizing the sum of access cost and monetary cost, maximizing the security-level agreement and trust among the community clouds. Furthermore, an efficient and comprehensive selection algorithm is devised to extract the best group of community clouds in *MC*³. Finally, the extensive simulation experiments and performance analysis of the proposed algorithm are conducted. The results demonstrate that the proposed algorithm outperforms the minimal set coverings based algorithm and the random algorithm. Moreover, the proposed comprehensive community clouds selection algorithm can guarantee good global performance in terms of access cost, monetary cost, security level and trust between user and community clouds.

Index Terms—Community Cloud, Collaboration, Task, *SPPR*, *STPPR*

1 INTRODUCTION

As a new computing paradigm, Cloud Computing has been widely used in many resource-intensive computational and commercial applications, such as Amazon EC2 [1] and Google AppEngine. Cloud Computing is a large-scale distributed computing paradigm driven by economies of scale, in which a pool of dynamically-scalable and visualized computing power, storage, platforms, and services are delivered on-demand to external customers over the Internet [2], [3]. Clouds can be classified into four categories: private cloud, public cloud, hybrid cloud, and community cloud, according to their

service types, deployment models and operation scale. The community cloud is an important social structure but also a challenging technology paradigm [19], providing the capability to realize business processes in the cloud and at the same time preserving a high security level by means of hybrid deployment models.

The ubiquitous nature of social networks and cloud computing facilitate the rapid development of community clouds which enable friends to share resources within the context of a social network [16], [17]. Social networks are used to reflect real-world relationships that allow users to share information, form connections between one another and essentially create dynamic virtual organizations. There are many instances of integration between social networks and cloud computing. For example, Facebook users can build the applications based on scalable cloud platform that are hosted by Amazon web service.

However, some challenging issues are arising along with these emerging developing paradigms, *e.g.*, efficient infrastructure for social collaboration in community clouds. Cloud collaboration is a newly emerging way of sharing and co-authoring computer files by virtue of cloud computing, whereby documents are uploaded to a central “cloud” for storage, where they can then be accessed by others. New cloud collaboration technologies have allowed users to upload, comment and collaborate on documents and even amend the document, evolving the document within the cloud. Meanwhile, businesses over the last few years have increasingly been switched to use of cloud collaboration ¹.

A number of studies on cloud collaboration have been reported recently [5], [12], [14], [15]. The cloud collaboration provides the following benefits for us: 1) simple storage and retrieve [20]; 2) efficient collaboration: allows multiple people to collaborate on content [13]; 3) complex collaboration: offers a complete environment including storage and retrieval, content management, and office productivity applications such as word processing, spreadsheet, presentations, calendaring, and workflow [6], [7]; 4) functional applications: offers functional appli-

¹School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China

²Department of Computing, University of Bradford, Bradford, UK

³Faculty of Engineering and Information Technology, University of Technology, Sydney, Australia

⁴Department of Computer Science, St. Francis Xavier University, Antigonish, Canada

*Corresponding author: L.T. Yang, Email:ltyang@gmail.com

1. http://en.wikipedia.org/wiki/Cloud_collaboration

cations, *e.g.*, project management; 5) business and social networking; offers the ability for social communities to be formed in addition to simple storage and retrieval of documents [18].

Different from the existing work considering clouds collaborative computing and community clouds computing only, this paper attempts to make collaboration among the community clouds and improve the computing performance for a complex task with the goal of the high economy-efficiency, communication-efficiency, secure, and trust. In this paper, we focus on the social collaborative computing among the community clouds but not within a certain cloud terminal in order to complete the given task and maximize the Security-based Trust-enabled Performance Price Ratio (*STPPR*) of collaborations.

The major contributions of this paper are summarized as follows:

- An efficient computational model for multi-community-cloud social collaboration (MC^3) model is formulated and described as a weighted bidirectional clique. Furthermore, upon this proposed computational model, the selection problem of community clouds is addressed by identifying the best group of community clouds with maximum *STPPR* of collaborations.
- We prove that the problem of finding a group of community clouds that maximizes the security-based trust-enabled performance price ratio is NP-hard. To guarantee the efficient and economical task–community cloud offloading decisions, a global optimal selection algorithm based on *STPPR* maximization, namely *Max-STPPR* is proposed to identify the best group of community clouds in MC^3 . Moreover, the scalability of the proposed algorithm is proposed by optimizing the data structure.
- The extensive simulations are conducted for evaluation of the proposed computational model. Experimental results demonstrate that the proposed algorithm reduces the sum of access cost and monetary cost as well as improves the security level and trust between user and community clouds in general. In particular, two devised comprehensive community clouds selection algorithms greatly guarantee the better global performance on aspects of access cost, monetary cost, security level and trust. Experimentally, the *Max-STPPR* algorithm is 21.5 times and 3.38 times better than the baseline algorithms: *random* algorithm and selection algorithm based on minimal set-covering (*Min-SC*) algorithm, in terms of *STPPR* with different number of community clouds, respectively; likewise, the *Max-STPPR* improves 290% and 81.1% on the basis of *random* and *Min-SC*-based algorithms in terms of *STPPR* with different size of task.

The rest of this paper is organized as follows. Section 2 surveys the related work on cloud collaboration. The ar-

chitecture of MC^3 and modules functions are described in Section 3. Section 4 focuses on the optimization process of community clouds selection in MC^3 in order to maximize the (*STPPR*) of collaborations. Then, the simulation experiments for validating the effectiveness and efficiency of the proposed algorithms and the performance results are presented in Section 5. Finally, Section 6 concludes this paper.

2 RELATED WORK

To complete a given complex task and improve the (*STPPR*) of collaborations, an optimal selection process for cloud computing partners / community clouds needs to consider the individual / community characteristics of each cloud computing environment in order to be effective. In the field of cloud collaborative computing, there are three main patterns for cloud collaborative computing, *i.e.*, *pooled cloud computing*, *sequential cloud computing* and *networked cloud computing* according to their structural properties [4].

There are two types of clouds, *i.e.*, *public cloud* and *community cloud*. Armbrust *et. al* [4] pointed out that one business actor handles a service from the central point which can be represented either by a public cloud environment or a community cloud environment in pooled cloud computing. The community clouds differ from public clouds because they are available only for the members of a specific community. The interdependencies of the specific community remain the same, which is the reason why both public clouds and community clouds belong to the same collaboration type [5]. From the security point of view, the community clouds have a higher security level than public clouds [8]. Suppose a given task is decomposed to several service flows. The sequential cloud computing is an available computing model to complete the given task. It contains public clouds and represents a chain of interdependencies by means of service flows (cloud chain). It has a low security level comparable to that of public clouds. Additionally, the networked cloud computing model is commonly used in industrial clouds where the output of each point can become the input for the other points [8]. However, the fact that they constitute a closed network of private clouds guarantees a high level of security. Obviously, the *sequential cloud computing* and *networked cloud computing* belong to multiple clouds social collaborative computing. However, we investigate the computational model for multi-community-cloud social collaboration (MC^3) where all of the community clouds are networked as a graph in this paper. The architecture and topology of MC^3 are described in the following section.

3 THE ARCHITECTURE OF MC^3

In this section, we first provide the preliminaries of heterogeneous social networks, community cloud, and tasks followed by a motivating scenario with MC^3 computing model, then the architecture of MC^3 is devised.

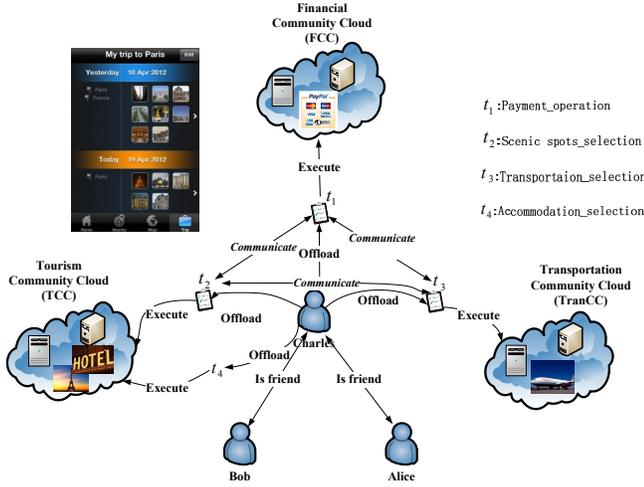


Fig. 1. A Tourism Scenario of Multi-Community Cloud Social Collaboration

3.1 Preliminaries

Definition 1: (Heterogeneous Social Networks) A heterogeneous social network $H = (V_H, E_H, L)$ is a directed labeled graph, where V_H is a set of nodes, L is a set of relation labels, and $E_H \subseteq V_H \times L \times V_H$ is a set of edges. Let a triple $\langle source, label, target \rangle$ be an edge, the function $type(E) = \{\{l_1, \dots, l_j\}, l_i \in L, j \geq 1\}$ maps each edge e_H into its set of typed labels. Note that the inverse edge set E^{-1} is the set of all edges (v, l_i^{-1}, u) .

Definition 2: (Community Cloud) A community cloud in computing is a collaborative effort in which infrastructure is shared between several organizations from a specific community with common concerns. A community cloud v is formalized as a 3-tuple $v = \{p_v, s_v, T_v\}$, where p_v is the unit service price for executing a task, s_v is the Security-Level Agreement (SLA) value, and T_v is the set of tasks which can be executed in community cloud v .

Definition 3: (Tasks) A Task τ consists of m sub-tasks, i.e., $\tau = \{t_1, t_2, \dots, t_m\}$ and the size of each sub-task is represented by $D(t_i)$ ($i = 1, 2, \dots, m$). Each sub-task must be offloaded and executed at one community cloud.

3.2 Motivating Scenario

The computational paradigm of multi-community-cloud social collaboration can be regarded as a heterogeneous social network $H = (V_H, E_H, L)$ which is a directed labeled graph. In this graph, V_H denotes the users, community clouds, and tasks; L contains "Is friend", "Offload", "Execute", and "Communicate"; E_H indicates the social relationships between users, execution between tasks and community clouds, and communication between tasks.

Consider an intelligent decision and recommendation service of travelling plan for tourists, a certain tourist

may launch his travelling plan request with the inputs including location, destination, departure time, ending time, and affordable price range by an application in his smart phone, then a detailed travelling plan will be recommended to tourists. As shown in Figure 1, a tourism scenario of the multi-community-cloud social collaboration paradigm is depicted. Clearly, there exist two reciprocal friendships between tourists *Charles* and *Bob*, *Charles* and *Alice*. One day, *Charles* plans to visit *Paris* for taking a vacation, he submits his travelling plan request τ via his smart phone. His request is actually further represented with several sub-tasks t_1, t_2, t_3, t_4 , i.e., $\tau = \{payment_operation, scenicspots_selection, transportation_selection, accommodation_selection\}$.

Then, those sub-tasks will be offloaded to various community clouds intelligently. In particular, t_1 is executed on the financial community cloud (FCC) for operating some necessary payment of travelling, t_2 and t_4 can be offloaded and executed on the tourism community cloud (TCC) for obtaining useful information about tourism destination and accommodation, and t_3 is executed on the transportation community cloud (TranCC) for making the reservations of tickets, such as flight, bus, ship tickets. However, the offloaded tasks have to communicate with each other in order to complete the given travelling plan task τ collaboratively. Actually, there exist different ways to complete the task because of diverse functionalities of community clouds. For example, TCC can execute t_1 for completing the payment and TranCC cloud also can execute t_4 for obtaining the accommodation information. In other words, a sub-task can be offloaded and executed on different community clouds. However, some constraints such as communication cost, security level, and price are taken into account for selecting the community clouds. Additionally, the trust relationships between tourists and community clouds, and between tourists are playing an important role in the aspect of community clouds selection. For instance, if *Charles's* friend *Alice* wants to go to *Paris* as well, this computing paradigm will directly recommend the travelling plan according to *Charles's* historical travelling plan to her instead of offloading tasks to the community clouds. From the energy-saving point of view, the multi-community-cloud social collaboration paradigm can reduce the costs and save the energy. In summary, we consider the social collaboration properties, SLA constraint, and trust constraint of the community cloud in this paper and propose a new computing paradigm: multi-community-cloud social collaboration, referred to as MC^3 .

3.3 The Proposed MC^3 Computing Architecture

The MC^3 arises from concerns over community cloud computing and social computing, specifically controlled by vendors and lack of environmental sustainability. First, MC^3 combines the use cases of community

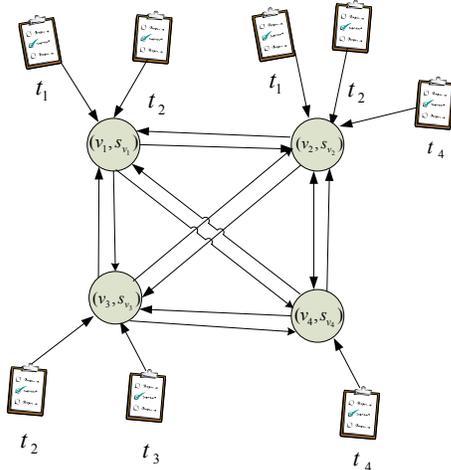


Fig. 2. The Topology of MC^3 Computing Paradigm

cloud computing, while making greater use of self-management advances from Autonomic Computing. Second, it replaces vendor community clouds by shaping the underutilized resources of user machines to form a Multi-Community-Cloud Collaboration Network, so called the topology of the MC^3 , as shown in Figure 2.

Definition 4: (Multi-Community-Cloud Collaboration Network) In a heterogeneous social network H , a multi-community-cloud collaboration network can be extracted and simplified as a weighted bidirectional clique, i.e., $\Omega = (V, E)$. A domain of sub-tasks is represented as $T = \{t_1, t_2, \dots, t_n\}$. Each vertex in V represents a community cloud that holds a set of virtual computing resources for $T_v \in T$ in the clique. Each edge $(v_i, v_j) \in E$ in Ω denotes the cost (e.g., access cost and monetary cost) that are caused by tasks communication between community clouds. s_{v_i} refers to the SLA constraint of community cloud v_i . A task $\tau = \{t'_1, t'_2, \dots, t'_m\} (m \leq n)$ required to accomplish the task, that is $\tau \subseteq T$.

Definition 5: (Feasible Group of Community Clouds) A feasible group of community clouds \tilde{V} for $\tau = \{t_1, t_2, \dots, t_m\}$ is defined as follows,

$$\tilde{V} = \{ \langle t_1, (v_{t_1}, s_{v_{t_1}}) \rangle, \dots, \langle t_m, (v_{t_m}, s_{v_{t_m}}) \rangle \} \quad (1)$$

where $v_{t_i} \in C(t_i)$ is a community cloud in V with the capability for executing sub-task t_i , and $s_{v_{t_i}}$ is the SLA value of community cloud v_{t_i} . $\langle t_i, (v_{t_i}, s_{v_{t_i}}) \rangle$ implies that community cloud v_{t_i} (with the SLA value of $s_{v_{t_i}}$) is in charge of executing sub-task t_i .

Let \tilde{V}_τ be a set of all feasible groups of community clouds, i.e., $\tilde{V}_\tau = \{\tilde{V}_1, \tilde{V}_2, \dots, \tilde{V}_w\}$ where w is the number of feasible group of community clouds.

For convenience, Table 1 lists important variables used throughout the paper.

As can be seen from Figure 2, a weighted 4-clique includes 4 community clouds v_1, v_2, v_3 , and v_4 with their corresponding SLA values $s_{v_1}, s_{v_2}, s_{v_3}$, and s_{v_4} as well as a complex task $\tau = \{t_1, t_2, t_3, t_4\}$. Note that each commu-

Variables	Descriptions
Ω	MC^3 computing paradigm
T	task domain
T_v	the set of virtual computing resources in community cloud v
τ	a complex task
k	the number of community clouds in Ω
m	the number of sub-tasks in τ
$C(t_i)$	the set of community clouds with computing resource t_i
SAC	sum of access cost
SMC	sum of monetary cost
SLA	security-level agreement
SPPR	security-based performance price ratio
STPPR	security-based trust-enabled performance price ratio
V	the set of community clouds in Ω
\tilde{V}	a feasible group of community clouds
\tilde{V}_τ	the set of all \tilde{V}
\hat{V}	the best group of community clouds
$D(t_i)$	the data size of task t_i
$p(v_i)$	the unit price of community cloud v_i
r_j^A	the trust value between user A and community cloud j
AC	access cost matrix
MC	unit monetary cost matrix
C	allocation matrix of computing capability

TABLE 1

Important Variables Used in The Paper.

nity cloud can execute different sub-tasks. For example, the community cloud v_1 is responsible to execute sub-tasks t_1 and t_2 . The community cloud v_2 is responsible to execute sub-tasks t_1, t_2 , and t_4 . Obviously, a certain sub-task can be executed in multiple community clouds. For example, t_2 can be executed in v_1, v_2 , and v_3 . In MC^3 , the output of each community cloud can become the input for the other community clouds. However, the fact that they constitute a closed network of community clouds guarantees a high level of security. In other words, the multiple community clouds collaborate each other to complete a given task τ which is composed of sub-tasks t_1, t_2, t_3 , and t_4 . From the perspective of both social collaborative characteristic and SLA constraints, how and which community clouds should be selected as the terminals of virtual computing resource and obtain the effective collaboration for a given task τ are facing with following challenges:

- **(Access Cost)** Once all the sub-tasks are offloaded onto the different community clouds, MC^3 requires their reciprocal communication to complete the given task collaboratively for minimizing the sum of access cost between tasks;
- **(Monetary Cost)** Due to the unit price of the community cloud resources, the overall monetary cost factor should be considered for multiple community clouds selection;
- **(SLA Constraint)** The SLA constraint of the community cloud should be taken into account, that is to maximize the overall Security-Level Agreement (SLA) in order to guarantee the security level of the tasks;
- **(Trust Constraint)** When a user makes the task offloading decisions (community clouds selecting strategy), we not only consider the above three

factors, but also the trust degree between him and the community cloud should be taken into account. For the multi-user case, the trust can be inferred by using the trust inference and social networks analysis approaches.

4 OPTIMIZE THE SELECTION PROCESS OF COMMUNITY CLOUDS

This section presents the formulation of the selection process of community clouds and discusses the affecting factors of this selection process. Based on the detailed analysis and discussion, two comprehensive community clouds selection schemes with the consideration of access cost, monetary cost, and SLA constraint as well as with the consideration of additional trust constraint are devised, respectively.

4.1 Problem Formulation

Problem Statement (Community Clouds Optimal Selection Process) For a given MC^3 computing paradigm $\Omega = (V, E)$ and a task $\tau = \{t_1, t_2, \dots, t_m\}$, the optimal selection process of community clouds problem ($MC^3_{opt.selection}$) is to extract a group of community clouds $\tilde{V} \in \tilde{V}_\tau \in V$ for τ from Ω so that the overall access cost and monetary cost are minimized, SLA values and the average trust degree of the selected community clouds are maximized.

The addressed problem is a complex multi-objective programming problem. First, we will analyze the impact factors associated with selection of community clouds. Then, two comprehensive selection schemes and corresponding algorithms with the consideration of the multi-objects together will be explored.

4.2 Affecting Factors for Community Clouds Selection

We discuss the impact factors including access cost, monetary cost, security-level agreement constraint and trust constraint for effective community clouds collaboration, respectively, in this section.

4.2.1 Access Cost

In an MC^3 computing paradigm $\Omega = (V, E)$, suppose a task $\tau = \{t_1, t_2, \dots, t_m\}$, these sub-tasks are offloaded onto a subset of community clouds $\tilde{V} \in V$ and $\tilde{V} = \{v_1, v_2, \dots, v_k\}$ according to the allocation matrix of computing capability C . C is a 0-1 matrix in which each element $C[i, j] = 1$ denotes the community cloud v_i has the computing capability of task t_j , then collaborated by communication between tasks. Let Tr be a task traffic matrix in which $Tr[t_i, t_j]$ denotes the traffic between task t_i and t_j . The sum of access cost (SAC) is defined as

$$SAC(\tilde{V}) = \sum_{i=1}^m \sum_{j=1}^m AC(v_i, v_j) * Tr[t_i, t_j], \quad (2)$$

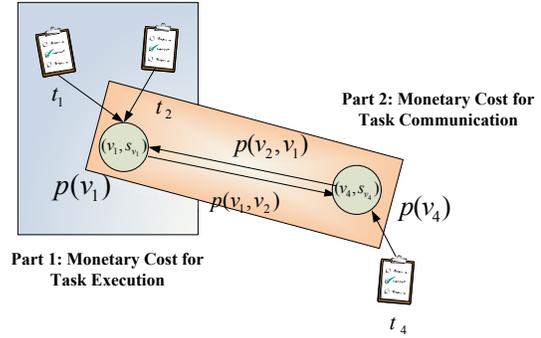


Fig. 3. The Monetary Cost for Task Execution and Communication

where $v_i \in C(t_i), v_j \in C(t_j), t_i \in T_{v_i}, t_j \in T_{v_j}$ and $AC(v_i, v_j)$ denotes the access costs from community cloud v_i to community cloud v_j . Obviously, $SAC(\tilde{V})$ depends on 1) the traffic between task t_i executed in community cloud v_i and t_j executed in community cloud v_j ; and 2) the access cost between the community clouds v_i and v_j .

4.2.2 Monetary Cost

From the monetary science point of view, a user attempts to select several affordable and cheap community clouds to execute offloaded tasks. In this section, an analysis of monetary cost for community clouds selection is investigated.

In an MC^3 computing paradigm $\Omega = (V, E)$, for a feasible group of community clouds $\tilde{V} \in V$ and $\tilde{V} = \{v_1, v_2, \dots, v_k\}$, and a task $\tau = \{t_1, t_2, \dots, t_m\}$. Figure 3 shows that the sum of monetary cost (SMC) for executing a given task including the two parts: 1) the monetary cost generated by community clouds themselves which is proportional to the data size of tasks that are offloaded on them, and 2) the monetary cost generated by the communication between tasks which is proportional to traffic between tasks. Hence, it is formalized as follows,

$$SMC(\tilde{V}) = \sum_{i=1}^m \sum_{j=1}^m (p(v_i)D(t_i) + p(v_i, v_j)Tr[t_i, t_j]), \quad (3)$$

where $v_i \in C(t_i), v_j \in C(t_j), t_i \in T_{v_i}, t_j \in T_{v_j}$; $p(v_i)$ denotes the unit service price of community cloud v_i per Gigabyte; $p(v_i, v_j)$ refers to the unit service price of communication between community clouds.

Clearly, the monetary cost is to evaluate the total expense for completing a given task.

4.2.3 Security-level Agreement Constraint

SLA constraint is an important factor for community clouds selection. In other words, a user prefers to select the community clouds with high overall SLA values for executing his task τ because of privacy protection purpose. For a feasible group of community clouds

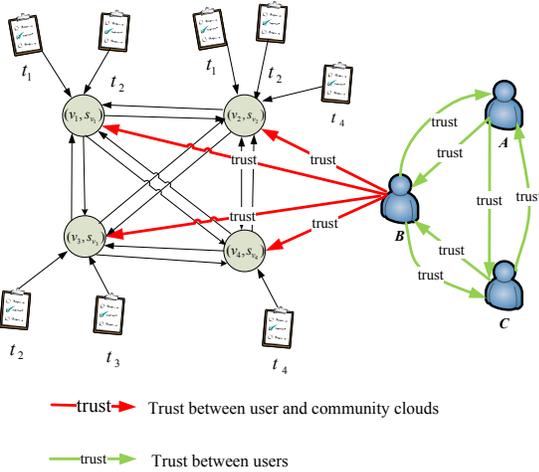


Fig. 4. The Trust Relationships in MC^3

$\tilde{V}_i = \{v_1, v_2, \dots, v_k\}$, the overall SLA value of V_i , i.e., $SLA(\tilde{V}_i)$ equals to the minimum SLA value among the community clouds $s_{v_i}, i = 1, \dots, k$ [23], [24]. Consider the Figure 4 as an example, for a feasible group of community clouds $\tilde{V} = \{v_1, v_3, v_4\}$, the $SLA(\tilde{V})$ can be obtained with following equation,

$$SLA(\tilde{V}) = \min\{s_{v_1}, s_{v_2}, s_{v_3}\} \quad (4)$$

4.2.4 Trust Constraint

For the community cloud users in a social network, the trust plays a key and potential role for tasks offloading. From the sociology point of view, users prefer to choose some community clouds which have been chosen by their friends for tasks offloading and computing because of their reciprocal trust. Figure 4 illustrates that the trust relationships exist in MC^3 with two forms: 1) trust between users and clouds; 2) trust between users. If the trust values are not available between some users and clouds, we can infer them by trust transition computing. If the trust values are available between a user i and a feasible group of community clouds \tilde{V}_i (i.e., r_1^i, \dots, r_k^i), we adopt the average trust value of the group of community clouds, $AT(\tilde{V}_i) = \frac{\sum_{j=1}^k r_j^i}{k}$, as the selection criteria.

Especially, if the trust values are not available between some users and community clouds, we adopt the trust maximization principle in social networks [21], [22] to infer them. For example, the trust values between user A and community clouds can be inferred with trust values between user B and community clouds through two trust transitive routes, i.e. $A \rightarrow B$ and $A \rightarrow C \rightarrow B$, then we select the best trust transitive route which can maximize the trust values between user A and community clouds, i.e.:

$$r_j^A = \text{Max}\left\{ \underbrace{r_j^B * r_B^A}_{\text{route: } A \rightarrow B}, \underbrace{r_j^C * r_C^A * r_B^C}_{\text{route: } A \rightarrow C \rightarrow B} \right\}, j = 1, 2, \dots, k, \quad (5)$$

where r_j^A denotes the trust value from user A to community cloud j , r_B^A indicates the trust value from user A to user B .

In this case, we have two selection strategies for the best group of community clouds :

- **(Strategy 1)** If current user will offload the tasks which are exactly the same as the tasks offloaded by a historical user who has trust values between him and community clouds, then the selection results can be directly recommended to current user. As depicted in Figure 4, if current user A will offload the tasks which are exactly the same with the tasks offloaded by user B , we will recommend the best group of community clouds of user B to user A instead of re-computing.
- **(Strategy 2)** If current user will offload the tasks which are not the same as the tasks offloaded by some historical users who have trust values between him and community clouds, then we calculate the trust values between current user and community clouds by Eq.(5) and further obtain the best group of community clouds by maximizing its average trust value.

4.3 The Proposed Comprehensive Selection Schemes

By analyzing the impact factors of community clouds selection, two proposed comprehensive community clouds selection schemes: 1) Security-based Performance Price Ratio Aware $MC^3_{opt.selection}$ and 2) Security-based Trust-enabled Performance Price Ratio Aware $MC^3_{opt.selection}$ are presented.

4.3.1 Security-based Performance Price Ratio Aware $MC^3_{opt.selection}$

This scheme is a single-user oriented multi-objective optimized selection algorithm of community clouds with the considerations of the access cost, monetary cost as well as the SLA constraint. Intuitively, an effective algorithm should achieve the minimal costs including access cost and monetary cost with the highest SLA value, namely security-based performance price ratio.

Definition 6: (Security-based Performance Price Ratio) In an MC^3 computing paradigm $\Omega = (V, E)$, for a group of community clouds $\tilde{V} \in V$ and $\tilde{V} = \{v_1, v_2, \dots, v_k\}$, the security-based performance price ratio ($SPPR$) is defined as

$$SPPR(\tilde{V}) = \frac{\alpha}{SAC(\tilde{V})} \times SLA(\tilde{V}) = \frac{\alpha SLA(\tilde{V})}{SAC(\tilde{V}) * SMC(\tilde{V})}, \quad (6)$$

where α is a constant for normalizing the value of $SPPR$. There exists a direct proportion relationship with SLA value and an inverse relationship with the multiplication of SAC and SMC. Therefore, improving the $SPPR(\tilde{V})$ is equivalent to reducing $SAC(\tilde{V}) * SMC(\tilde{V})$ and increasing $SLA(\tilde{V})$ simultaneously.

Under this constraint, the best group of community clouds selection process is formalized as follows,

$$\hat{V} := \arg \tilde{V}_i \in \tilde{V}_\tau \quad \max SPPR(\tilde{V}_i). \quad (7)$$

4.3.2 Security-based Trust-enabled Performance Price Ratio Aware $MC^3_{opt.selection}$

In a multi-user multi-community-cloud social collaboration model, trust factor should be taken into account. This section devotes to presenting a security-based trust-enabled performance price ratio aware $MC^3_{opt.selection}$ algorithm that integrates all challenges together. Then, we devise a novel global evaluation metric, namely security-based trust-enabled performance price ratio ($STPPR$) to rank the community clouds and output a best group of community clouds which can satisfy the requirements of the given task as well as maximize the $STPPR$ value.

Definition 7: (Security-based Trust-enabled Performance Price Ratio) In an MC^3 computing paradigm $\Omega = (V, E)$, for a group of community clouds $\tilde{V} \in V$ and $\tilde{V} = \{v_1, v_2, \dots, v_k\}$, the security-based trust-enabled performance price ratio ($STPPR$) is defined as

$$\begin{aligned} STPPR(\tilde{V}) &= \frac{\alpha}{SAC(\tilde{V})} \times SLA(\tilde{V}) \times AT(\tilde{V}) \\ &= \frac{\alpha SLA(\tilde{V}) AT(\tilde{V})}{SAC(\tilde{V}) * SMC(\tilde{V})}, \end{aligned} \quad (8)$$

where α refers to a constant which is used for normalizing the value of $STPPR$. Similar to the definition of $SPPR$, there exists a direct proportion relationship with SLA value, average trust (AT) value, and an inverse relationship with the multiplication of SAC and SMC . Therefore, improving the $STPPR(\tilde{V})$ is equivalent to reducing $SAC(\tilde{V}) * SMC(\tilde{V})$ and increasing $SLA(\tilde{V})$, $AT(\tilde{V})$ simultaneously.

Similarly, the best group of community clouds selection process is formalized as follows,

$$\hat{V} := \arg \tilde{V}_i \in \tilde{V}_\tau \quad \max STPPR(\tilde{V}_i). \quad (9)$$

Generally, the security-based trust-enabled performance price ratio aware $MC^3_{opt.selection}$ is a comprehensive scheme for multi-community-cloud selection. Because it takes access cost, monetary cost, security and trust into account, the following algorithm will be devised and elaborated upon this scheme.

Theorem 1: The $MC^3_{opt.selection}$ problem is NP-hard. The selection process of community clouds in $MC^3_{opt.selection}$ problem can be viewed as a special case of an existing NP-hard problem about seed nodes mining in Influence Maximization [9], [10]. In other words, a subset $\tilde{V} \in V$ with cardinality m of nodes (corresponding to seed nodes) is expected to extract such that maximizing the security-based trust-enabled performance price ratio (corresponding to influence coverage) in $MC^3_{opt.selection}$ problem. Therefore, the $MC^3_{opt.selection}$ problem is NP-hard as well.

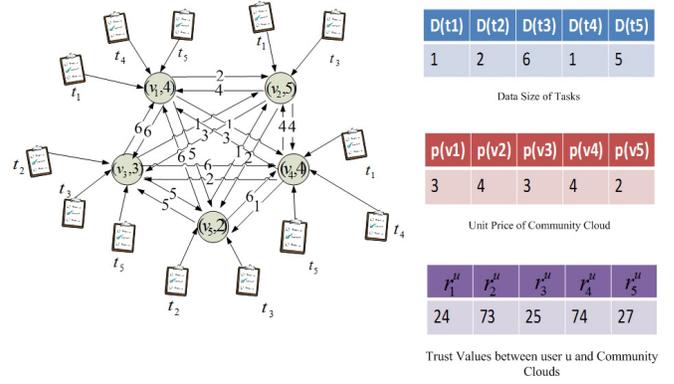


Fig. 5. An Instance of $MC^3_{opt.selection}$

Remark: The trust values between user u and community clouds are $r_1^u = 24$, $r_2^u = 73$, $r_3^u = 25$, $r_4^u = 74$, $r_5^u = 27$

To better understand the aforementioned affecting factors and the proposed comprehensive selection schemes, we take Figure 5 as an example of $MC^3_{opt.selection}$ problem to explain the working process of $MC^3_{opt.selection}$. The MC^3 computing paradigm is modeled as a weighted bidirectional 5-clique. The nodes v_1, v_2, v_3, v_4, v_5 in this weighted bidirectional 5-clique denote the five community clouds which hold the virtual computing resource of $\{t_1, t_4, t_5\}$, $\{t_1, t_3\}$, $\{t_2, t_3, t_5\}$, $\{t_1, t_4, t_5\}$, $\{t_2, t_3\}$, and have the SLA values $s_{v_1}=4$, $s_{v_2}=5$, $s_{v_3}=3$, $s_{v_4}=4$ and $s_{v_5}=2$, respectively. The data size of these tasks are $|t_1|=1$, $|t_2|=2$, $|t_3|=6$, $|t_4|=1$, $|t_5|=5$ (unit: Petabyte). The unit access cost matrix \mathbf{AC} between two community clouds is shown as follows,

$$\mathbf{AC} = \begin{pmatrix} 0 & 2 & 6 & 1 & 5 \\ 4 & 0 & 3 & 4 & 2 \\ 6 & 1 & 0 & 6 & 5 \\ 3 & 4 & 2 & 0 & 1 \\ 6 & 1 & 5 & 6 & 0 \end{pmatrix}$$

Obviously, the access cost matrix is an symmetry matrix. $AC[i, j]$ refers to the unit access cost between a task at community clouds i and another task at community cloud j . In particular, we assume there is no access cost among tasks at the same community cloud. Therefore, the diagonal entries in \mathbf{AC} are "0".

Let the task traffic matrix \mathbf{Tr} between tasks is as follows,

$$\mathbf{Tr} = \begin{pmatrix} 0 & 5 & 3 & 4 & 3 \\ 1 & 0 & 0 & 1 & 5 \\ 4 & 5 & 0 & 4 & 2 \\ 0 & 1 & 0 & 0 & 5 \\ 3 & 4 & 2 & 3 & 0 \end{pmatrix}$$

Suppose the service prices of community clouds are $p(v_1)=3$, $p(v_2)=4$, $p(v_3)=3$, $p(v_4)=3$, $p(v_5)=2$ (unit:USD/PB); The unit monetary cost matrix \mathbf{MC} between the community clouds,

$$\mathbf{MC} = \begin{pmatrix} 0 & 3 & 2 & 3 & 1 \\ 5 & 0 & 4 & 5 & 4 \\ 2 & 3 & 0 & 2 & 6 \\ 5 & 6 & 4 & 0 & 3 \\ 1 & 2 & 1 & 1 & 0 \end{pmatrix}$$

In unit monetary cost matrix MC , each entry $MC[i, j]$ denotes the expense for collaborating between the tasks on community cloud i and tasks on community cloud j . Note that, if tasks on the same community cloud, the monetary cost for their collaboration is zero, i.e., $MC[i, i] = 0$.

Figure 5 illustrates an instance of $MC_{opt.selection}^3$ with nodes indicating the community clouds, directed edges indicating the access costs from source community cloud to sink community cloud. For example, $AC(v_2, v_3) = 3$ denotes the unit access cost from community cloud v_2 to community cloud v_3 . The optimal selection process is elaborated as follows: In MC^3 computing paradigm, to complete a given task $T = \{t_1, t_2, t_3, t_4, t_5\}$, we need to select the sub-sets of community clouds which cover all sub-tasks and make them to collaborate each other. This process is exactly same with the problem of finding set coverings. In Figure 5, we can find 15 feasible groups of community clouds that satisfy our task execution requirement. However, we can easily obtain the optimal group of community clouds and corresponding tasks assignment schemes in terms of the minimal sum of access cost, minimal monetary cost, maximal SLA, maximal SPPR, maximal STPPR as follows:

- Access Cost: the group of community clouds with minimal SAC value is $\hat{V} = \{v_2, v_3, v_4\}$ and the corresponding task assignment scheme is to offload the tasks t_1 on v_2 , $\{t_2, t_3, t_5\}$ on v_3 and t_4 on v_4 . The $SAC(\hat{V})$ is 117.
- Monetary Cost: the group of community clouds with minimal SMC value is $\hat{V} = \{v_1, v_5\}$ and the corresponding task assignment scheme is to offload the tasks $\{t_1, t_4, t_5\}$ on v_1 and $\{t_2, t_3\}$ on v_5 . The $SMC(\hat{V})$ is 69.
- Security-level Agreement Constraint: the group of community clouds with maximal SLA value is $\hat{V} = \{v_1, v_2, v_3\}$ and the corresponding task assignment scheme is to offload the tasks $\{t_1, t_4, t_5\}$ on v_1 , t_3 on v_2 and t_2 on v_3 . The $SLA(\hat{V})$ is 3.
- Trust Constraint: the group of community clouds with maximal average trust value is $\hat{V} = \{v_2, v_4, v_5\}$ and the corresponding task assignment scheme is to offload the tasks $\{t_3\}$ on v_2 , t_1, t_4, t_5 on v_4 and t_2 on v_5 . The $AT(\hat{V})$ is 64.4.
- Security-based Performance Price Ratio Aware $MC_{opt.selection}^3$: the best group of community clouds is $\hat{V} = \{v_1, v_5\}$ and the corresponding task assignment scheme is to offload the tasks t_1, t_4, t_5 on v_1 and t_2, t_3 on v_5 . The $SPPR(\hat{V})$ is 32.75.
- Security-based Trust-enabled Performance Price Ratio Aware $MC_{opt.selection}^3$: the best group of com-

munity clouds is $\hat{V} = \{v_4, v_5\}$ and the corresponding task assignment scheme is to offload the tasks t_1, t_4, t_5 on v_4 and t_2, t_3 on v_5 . The $STPPR(\hat{V})$ is 1780.36.

4.4 Algorithm Description

Based on the above example and optimization process of community clouds selection, it is clear that our goal is to find the best group of community clouds in MC^3 in order to optimize the above constraints. This section presents our algorithm for finding the best group of community clouds.

Since Problem 1 $MC_{opt.selection}^3$ is an NP-hard problem, we may address this problem with greedy algorithm or 2-approximation approach proposed in [11]. However, these approximate algorithms cannot guarantee the global optimal solutions so that the tasks cannot be effectively and economically offloaded and executed in the community clouds. We hereby devise and employ a global optimal algorithm that finds the best group of community clouds. Algorithm 1 presents the pseudo code of the proposed algorithm in this paper. It takes an MC^3 computing paradigm Ω with a set of community clouds $V = \{v_1, v_2, \dots, v_k\}$ and a task $\tau = \{t_1, t_2, \dots, t_m\}$ representing sub-tasks as the input. In addition, the parameter α is needed to set up at the beginning stage of the algorithm². The algorithm returns the best group of community clouds \hat{V} .

The working process of Algorithm 1 is described as follows. It first initializes \tilde{V}_τ to \emptyset . Then, the *SetcoveringsGenerator* algorithm as shown in Algorithm 2 is invoked to generate all feasible groups of community clouds \tilde{V}_τ which can meet the computational requirement of the given task τ (Lines 2–4). For each feasible group of community clouds $\tilde{V}_i \in \tilde{V}_\tau$, the following steps are performed. Lines 8–11 are to select the best group of community clouds with the considerations of SPPR. Lines 12–14 provide the entries for selecting the best group of community clouds by maximizing STPPR. Finally, it returns the best group of community clouds \hat{V} in terms of different optimization goals.

4.5 Algorithm Discussion

At the beginning of Algorithm 1, it stores all feasible group of community clouds with a linked list. Then, this algorithm goes to selection phrases (Lines 7–15) for obtaining the best group of community clouds according to different constraints. To further increase the scalability of the proposed algorithm, we optimize the algorithm as follows:

- 1) The algorithm recursively generates the feasible solutions, and just stores the current feasible solution and the best solution in the nodes instead of generating the linked list.
2. $\alpha=200000$ is set in the experiment of this paper

Algorithm 1 Finding the Best Group of Community Clouds**Input:**

An MC^3 computing paradigm, Ω ;
 Set of Community Clouds, $V = \{v_1, v_2, \dots, v_k\}$;
 A task, $\tau = \{t_1, t_2, \dots, t_m\}$;
 Parameter α ;
 Algorithms ID M ;
 1: Security-based Performance Price Ratio aware Selection (*Max-SPPR*);
 2: Security-based Trust-enabled Performance Price Ratio aware Selection (*Max-STPPR*);

Output:

The best group of community clouds \hat{V}

```

1: Initialize  $\tilde{V}_\tau = \emptyset$ 
2: begin
3:    $\tilde{V}_\tau = \tilde{V}_\tau \cup \text{SetcoveringsGenerator}(\tau, V)$ 
4: end
5: for  $i=0$  to  $|\tilde{V}_\tau|$  do
6:   begin
7:     switch( $M$ )
8:       begin
9:         case 1:
10:           $\hat{V} = \arg \tilde{V}_i \in \tilde{V}_\tau \max SPPR(\tilde{V}_i)$ 
11:         break;
12:        case 2:
13:           $\hat{V} = \arg \tilde{V}_i \in \tilde{V}_\tau \max STPPR(\tilde{V}_i)$ 
14:         break;
15:        end
16:   Return  $\hat{V}$ 
17: end

```

- 2) This algorithm compares the node with the current feasible solution to the node with the best solution in terms of various constraints and decides whether replacing the best solution node using current feasible solution.
- 3) At last, the finalized result is stored in the best solution node.

The major advantages of the refined algorithm are that it can execute the optimal selection with any scale within the promising period and guarantee the global optimal solution.

5 SIMULATION AND PERFORMANCE EVALUATION

In this section, the extensive simulation experiments of community clouds optimal selection in MC^3 are presented. We evaluate and compare the proposed approach with the baseline work in terms of the following performance metrics: access cost, monetary cost, SLA value, trust, security-based performance price ratio and security-based trust-enabled performance price ratio among different numbers of community clouds with various sub-tasks (*i.e.*, size of task).

Algorithm 2 SetcoveringsGenerator (τ, V)**Input:**

Set of Community Clouds, $V = \{v_1, v_2, \dots, v_k\}$;
 A task, $\tau = \{t_1, t_2, \dots, t_m\}$;

Output:

Setcoverings \tilde{V}_τ ;

```

1:  $\tilde{V}_\tau \leftarrow \emptyset$ 
2: while  $|\tilde{V}_\tau| \neq \prod_{i=1}^m |C(t_i)|$ 
3:    $U \leftarrow \tau$ 
4:    $SCs \leftarrow \emptyset$ 
5:   while  $U \neq \emptyset$ 
6:     do permutational select a  $v_i \in V$  that  $|T_{v_i} \cap \{t_i\}| \neq \emptyset$ 
7:      $U \leftarrow U - \{t_i\}$ 
8:      $SCs \leftarrow SCs \cup v_i$ 
9:   if  $SCs \in \tilde{V}_\tau$  then
10:     $\tilde{V}_\tau \leftarrow SCs \cup \tilde{V}_\tau$ 
11: return  $SCs$ 

```

5.1 Setup

Since the proposed computational model and algorithm are generic to any data of multi-community-cloud social collaboration, the simulation data we used in this paper will not affect the selection process of the proposed algorithm. Without loss of generality, considering the difficulties of obtaining the concrete data of multi-community-cloud social collaboration, we generate the simulation data with targeted pseudo-random numbers method, and then run the following set of algorithms and compare their effectiveness as well as efficiency of community clouds selection.

- **Minimal set covering based algorithm:** The basic idea of *Min-SC* algorithm is to find the groups of community clouds with the smallest cardinality of them that can complete the task τ in MC^3 . If we find more than one best groups of community clouds, such as $\hat{V}_1, \hat{V}_2, \dots, \hat{V}_k$, then we calculate the average value of the sum of access cost, the sum of monetary cost, the sum of SLA values, the sum of security-based performance price ratio and the sum of security-based trust-enabled performance price ratio among them, *i.e.*, $\frac{\sum_i^k (SAC(\hat{V}_i))}{k}$, $\frac{\sum_i^k (SMC(\hat{V}_i))}{k}$, $\frac{\sum_i^k (SLA(\hat{V}_i))}{k}$, $\frac{\sum_i^k (SPPR(\hat{V}_i))}{k}$ and $\frac{\sum_i^k (STPPR(\hat{V}_i))}{k}$ then claim them as the overall evaluation metric.
- **Random algorithm:** As a baseline comparison, we simply and randomly find some groups of community clouds that can complete the task T in MC^3 .
- **Access cost aware algorithm:** This algorithm *Min-SAC* targets to extract the group of community clouds with the minimal sum of access cost, *i.e.*, $\hat{V} := \arg \tilde{V}_i \in \tilde{V}_\tau \min SAC(\tilde{V}_i)$.
- **Monetary cost aware algorithm:** This algorithm *Min-SMC* targets to find the group of community clouds with the minimal monetary cost, *i.e.*, $\hat{V} := \arg \tilde{V}_i \in \tilde{V}_\tau \min SMC(\tilde{V}_i)$.

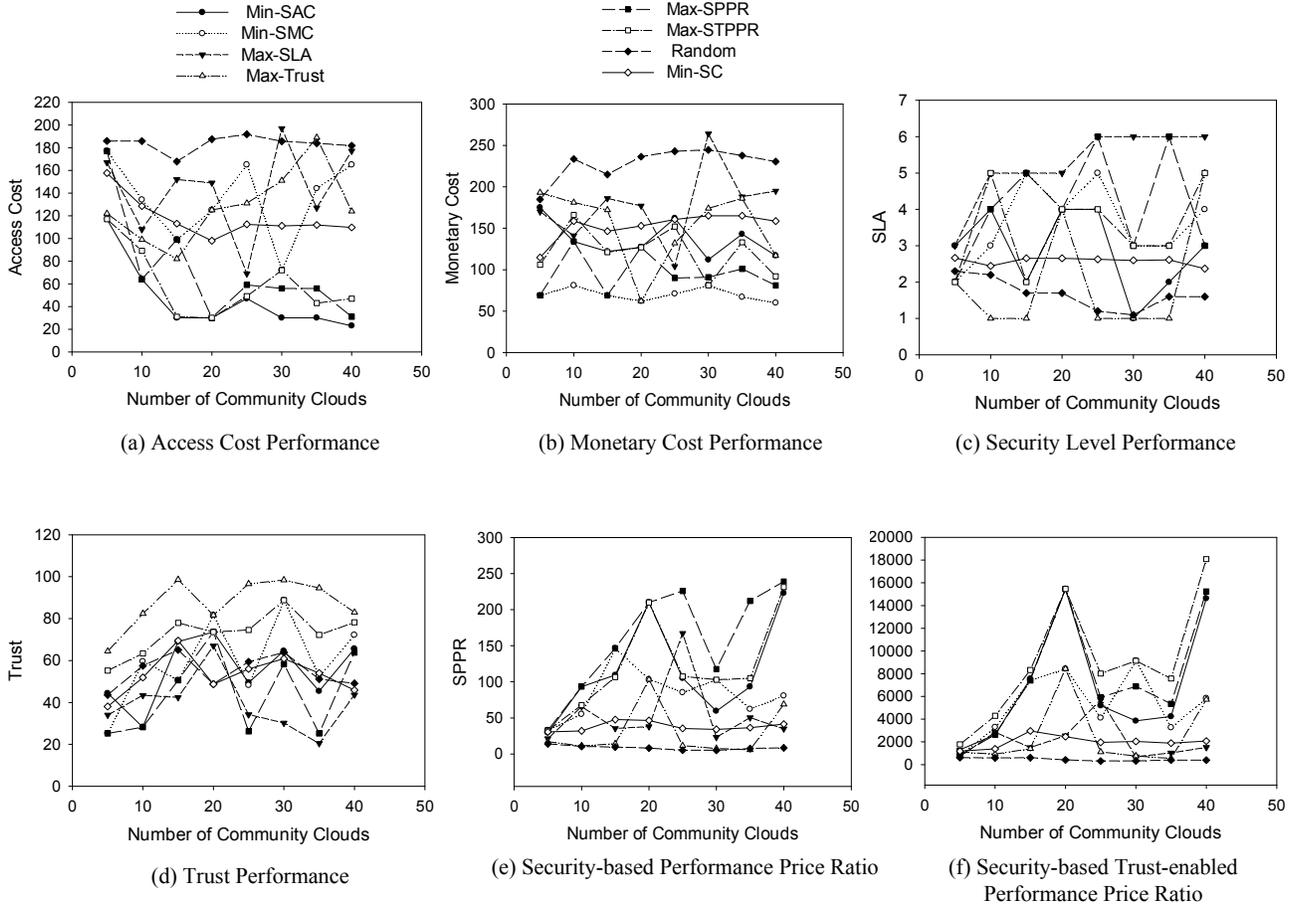


Fig. 6. The Effectiveness of Various Algorithms when Adjusting the Number of Community Clouds

- **Security-based algorithm:** This algorithm *Max-SLA* targets to identify the group of community clouds with the maximized SLA value, *i.e.*, $\hat{V} := \arg \tilde{V}_i \in \tilde{V}_\tau \max SLA(\tilde{V}_i)$.
- **Trust-based algorithm:** This algorithm *Max-Trust* targets to identify the group of community clouds with the maximized average trust value, *i.e.*, $\hat{V} := \arg \tilde{V}_i \in \tilde{V}_\tau \max AT(\tilde{V}_i)$.
- **Security-based performance price ratio aware algorithm :** This algorithm *Max-SPPR* combines the access cost, monetary cost as well as the SLA together for selecting the community clouds as the best group of community clouds.
- **Security-based trust-enabled performance price ratio aware algorithm:** This algorithm *Max-STPPR* combines the access cost, monetary cost, SLA as well as the trust together for selecting the community clouds as the best group of community clouds.

To obtain the effectiveness of the above algorithms on aspects of access cost, monetary cost, security level, trust, security-based performance price ratio and security-based trust-enabled performance price ratio. For sake of visual scalization, the parameter $\alpha=200000$ is set up at the initial stage of the algorithm. First, we fix the number of community clouds and obtain the

comparison results with the size of tasks ranging from 5 to 27 with step width=2. Then, we fix the size of tasks and get the comparison results by adjusting the number of community clouds from 5 to 40 with step width=5. From the efficiency point of view, we compare them by evaluating their corresponding running time, respectively.

5.2 Effect of the number of community clouds

Figures 6(a)–6(f) depict the performance comparison results of various algorithms in terms of various performance metrics. Obviously, the proposed algorithms can guarantee the requirement of the performance metric. More importantly, the proposed *Max-SPPR* and *Max-STPPR* algorithms are globally better than other algorithms. Especially, the *Max-STPPR* algorithm is 21.5 times and 3.38 times better than *random* and *Min-SC* algorithms in terms of *STPPR*, respectively. Therefore, it is an accepted global community clouds selection scheme with the considerations of access cost, monetary cost and security level for single user case. For multi-user case, the *Max-STPPR* algorithm achieves the satisfactory results with the considerations of access cost, monetary cost, security level and trust.

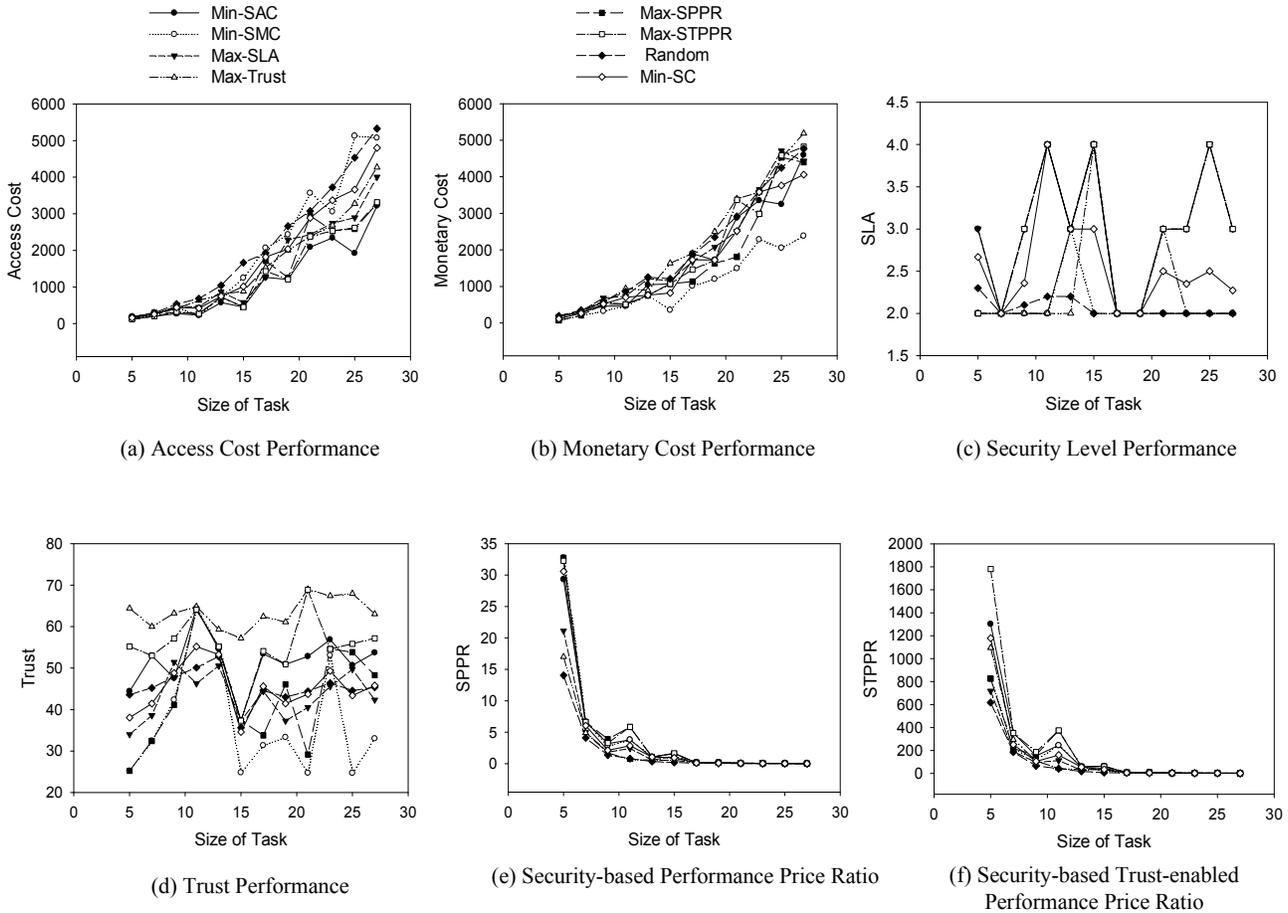


Fig. 8. The Effectiveness of Various Algorithms when Adjusting the Size of Task

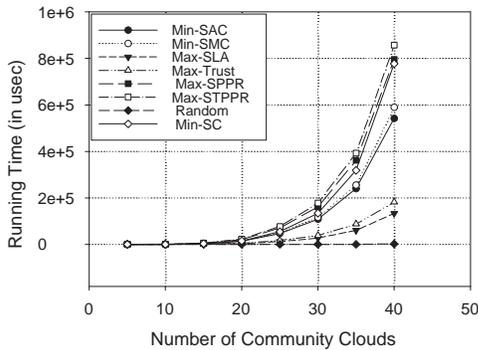


Fig. 7. The Running Time of Various Algorithms when Adjusting the Number of Community Clouds

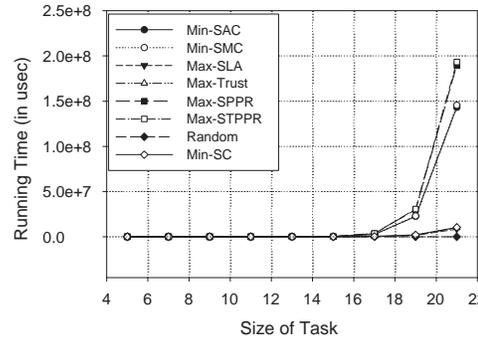


Fig. 9. The Running Time of Various Algorithms when Adjusting the Size of Task

5.3 Effect of the size of task

This section reports the performance comparison results of various algorithms when the size of task is adjusted. Clearly, Figures 8(a)–8(f) demonstrate that the proposed algorithms can guarantee the requirement of the performance metric. In particular, the proposed *Max-SPPR* and *Max-STPPR* algorithms are globally better than other algorithms, *i.e.*, the *Max-STPPR* improves 290% and 81.1% on the basis of *random* and *Min-SC* algorithms in

terms of *STPPR*. Therefore, it is an accepted global community clouds selection scheme with the considerations of access cost, monetary cost and security level for single user case. For multi-user case, the *Max-STPPR* algorithm achieves the satisfactory results with the considerations of access cost, monetary cost, security level and trust.

Algorithm	Time complexity
Min-SAC	$\Theta((\prod_{i=1}^m C(t_i)) \times m \times m + m \times m)$
Min-SMC	$\Theta((\prod_{i=1}^m C(t_i)) \times m \times m + m \times m)$
Max-SLA	$\Theta((\prod_{i=1}^m C(t_i)) \times m + m \times m)$
Max-Trust	$\Theta((\prod_{i=1}^m C(t_i)) \times m + m \times m)$
Max-SPPR	$\Theta((\prod_{i=1}^m C(t_i)) \times m \times m + m \times m)$
Max-STPPR	$\Theta((\prod_{i=1}^m C(t_i)) \times m \times m + m \times m)$
Random	$\Theta(m \times m)$
Min-SC	$\Theta((\prod_{i=1}^m C(t_i)) \times (m+k) + m \times m)$

TABLE 2

Time Complexity Analysis of Various Algorithms

5.4 Time Complexity Analysis

This section discusses the time complexity of the proposed multi-community-cloud selection algorithms. The steps of time complexity analysis are as follows,

- 1) **The time complexity of selecting the best group of community clouds:** The time complexity of selecting the best group of community clouds with algorithms *Min-SAC*, *Min-SMC*, *Max-SPPR*, *Max-STPPR* is $\Theta((\prod_{i=1}^m |C(t_i)|) \times m \times m)$; However, the time complexity of selecting the best group of community clouds with algorithms *Max-SLA*, *Max-Trust* is $\Theta((\prod_{i=1}^m |C(t_i)|) \times m)$;
- 2) **The time complexity of calculating the comparison results:** The time complexity of calculating the comparison results is $\Theta(m \times m)$;
- 3) **The time complexity of Random Algorithm:** The time complexity of this algorithm equals to the values of performance metrics calculation $\Theta(m \times m)$;
- 4) **The time complexity of Min-SC Algorithm:** The time complexity of calculating the values of performance metrics is $\Theta((\prod_{i=1}^m |C(t_i)|) \times (m+k) + m \times m)$.

To sum up, the time complexity of algorithms is the summation of the above two parts (*i.e.*, 1)+2)) as shown in Table 2.

As can be seen from Table 2, Figure 7, and Figure 9, the time complexity of two comprehensive algorithms *Max-SPPR* and *Max-STPPR* is exactly the same as that of algorithms *Min-SAC* and *Min-SMC* which only consider the access cost and monetary cost. But, the algorithms *Max-SPPR* and *Max-STPPR* cost much time than algorithms *Max-SLA*, *Max-Trust* and *Random*. The reason for this is that two comprehensive algorithms *Max-SPPR* and *Max-STPPR* take more factors into account when selecting the community clouds in MC^3 .

6 CONCLUSION

In this paper, we propose a novel cloud collaborative computing architecture MC^3 -Multi-Community-Cloud Social Collaboration Model. In addition, we formulate the problem of finding a group of community clouds that maximizes the security-based trust-enabled performance price ratio by analyzing the affecting factors for the selection process. Then, we provide a global optimal selection strategy of community clouds in MC^3 in order to improve the security-based trust-enabled performance

price ratio. The simulation results indicate that our algorithm reduces the sum of access cost, sum of monetary cost and improves the security level as well as trust between user and community clouds significantly compared with baseline selection algorithms. In particular, two devised comprehensive community clouds selection algorithms greatly guarantee the better global performance in terms of various optimized constraints. It is expected that the MC^3 computational model benefits to the complex tasks execution in multi-community-cloud platform.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China under Grant 61201219.

REFERENCES

- [1] Amazon.com: "Amazon Web Services (AWS)", <http://aws.amazon.com>, 2009.
- [2] I. Foster, Y. Zhao, I. Raicu and S. Lu: "Cloud Computing and Grid Computing 360-Degree Compared", In *Proc. of IEEE Grid Computing Environments Workshop (GCE)*, pages 1–10, 2008.
- [3] C. Wang, Q. Wang, K. Ren, N. Cao and W. Lou: "Toward Secure and Dependable Storage Services in Cloud Computing", *IEEE Transactions on Services Computing*, vol.5, no.2, pages 220–232, 2012.
- [4] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica and M. Zaharia: "A View of Cloud Computing", *Communications of the ACM*, vol. 53, no. 4, pages 50–58, 2010.
- [5] B. Martens, N. Zarvi, F. Teuteberg and O. Thomas: "Designing a Risk-based Partner Selection Process for Collaborative Cloud Computing Environments", In *Proc. of the 4th International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA)*, pages 237–242, 2011.
- [6] D.D. Snchez, F.A. Mendoza, A. Marln, D. Proserpio and P.A. Cabarcos: "Media cloud: An Open Cloud Computing Middleware for Content Management", *IEEE Transactions on Consumer Electronics*, vol.57, no.2, pages 970–978, 2011.
- [7] Z. Wei, G. Pierre and C. Chi: "CloudTPS: Scalable Transactions for Web Applications in the Cloud", *IEEE Transactions on Services Computing*, vol.5, no.4, pages 525–539, 2012.
- [8] T.W. Wlodarczyk, C. Rong and K.A.H. Thorsen: "Industrial Cloud: Toward Inter-enterprise Integration", In *Proc. of International Conference On Cloud Computing Technology And Science (CloudCom)*, pages 460–471, 2009;
- [9] D. Kempe, J. M. Kleinberg and E. Tardos: "Maximizing the Spread of Influence through a Social Network", In *Proc. of ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 137–146, 2003.
- [10] W. Chen, Y. Wang and S. Yang: "Efficient Influence Maximization in Social Networks", In *Proc. of ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 199–208, 2009.
- [11] M. Kargar and A. An: "Discovering Top-k Teams of Experts with/without a Leader in Social Networks, In *Proc. of ACM International Conference on Information and Knowledge Management(CIKM)*, pages 985–994, 2011.
- [12] F. Dolitzscher, A. Sulistio, C. Reich, H. Kuijs and D. Wolf: "Private cloud for collaboration and e-Learning services: from IaaS to SaaS", *Computing*, vol.91, no.1, pages 23–42, 2011.
- [13] J. Li, B. Li, Z. Du and L. Meng: "CloudVO: Building a Secure Virtual Organization for Multiple Clouds Collaboration", In *Proc. of ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, pages 181–186, 2010.
- [14] L. Popa, G. Kumar, M. Chowdhury, A. Krishnamurthy, S. Ratnasamy and I. Stoica: "FairCloud: sharing the network in cloud computing", In *Proc. of SIGCOMM*, pages 187–198, 2012.

- [15] B. Solomon, D. Ionescu, C. Gadea, S. Veres, M. Litoiu and Joanna Ng: "Distributed clouds for collaborative applications", In *Proc. of International Conference on Collaboration Technologies and Systems (CTS)*, pages 218–225, 2012.
- [16] K. Chard, S. Caton, O. Rana and K. Bubendorfer: "Social Cloud: Cloud computing in social networks", In *Proc. of IEEE International Conference on Cloud Computing (IEEE Cloud)*, pages 99–106, 2010.
- [17] W. Zhenyu, Z. Chunhong, J. Yang and W. Hao: "Towards Cloud and Terminal Collaborative Mobile Social Network Service", In *Proc. of IEEE International Conference on Social Computing (Social-Com)*, pages 623–629, 2010.
- [18] A.M. Thaufeeg, K. Bubendorfer and K. Chard: "Collaborative eResearch in a Social Cloud", In *Proc. of IEEE International Conference on E-Science (e-Science)*, pages 224–231, 2011.
- [19] A. Marinos and G. Briscoe: "Community cloud computing" In *Proc. of International Conference on Cloud Computing (CloudCom)*, pages 472–484, 2009.
- [20] J. Spillner, J. Miller and A. Schill: "Creating optimal cloud storage systems", *Future Generation Computer Systems*, vol.29, no.4, pages 1062–1072, 2013.
- [21] J. Zhan and X. Fang: "Trust Maximization in Social Networks", In *Proc. of International Conference on Social Computing, Behavioral-cultural Modeling and Prediction (SBP)*, pages 205–211, 2011.
- [22] G. Liu, Y. Wang, M.A. Orgun and E. Lim: "Finding the Optimal Social Trust Path for the Selection of Trustworthy Service Providers in Complex Social Networks," *IEEE Transactions on Services Computing*, vol.6, no.2, pages 152–167, 2013.
- [23] T. Ludescher, T. Feilhauer, and P. Brezany: "Security Concept and Implementation for a Cloud Based E-science Infrastructure", In *Proc. of ARES*, pages 280–285, 2012.
- [24] Y. Chen, Q. Shen, P. Sun, Y. Li, Z. Chen, and S. Qing: "Reliable Migration Module in Trusted Cloud Based on Security Level-Design and Implementation", In *Proc. of IPDPS Workshops*, pages 2230–2236, 2012.



Geyong Min is a Professor of Computer Science in the Department of Computing at the University of Bradford, United Kingdom. He received the PhD degree in Computing Science from the University of Glasgow, United Kingdom, in 2003, and the B.Sc. degree in Computer Science from Huazhong University of Science and Technology, China, in 1995. His research interests include Next Generation Internet, Wireless Communications, Multimedia Systems, Information Security, Ubiquitous Computing, Modelling and Performance Engineering. His recent research has been supported by UK EPSRC, Royal Society, Nuffield Foundation, and European FP. Prof. Min has published over 200 research papers in prestigious international journals, including IEEE Transactions on Computers, IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Communications, IEEE Transactions on Wireless Communications, and IEEE Transactions on Multimedia, and in reputable international conferences, such as ICDCS, IPDPS, GLOBECOM, and ICC. He is an Editorial Board member of 9 international journals. He served as the Guest Editor for 17 International Journals and was the Chair or Vice-Chair of 30 international conferences/workshops. He was awarded the Outstanding Leadership Awards from IEEE International conferences C-IT2010/ScalCom2010/ICESS2010, ScalCom2009, HPCC2008. Contact him at g.min@brad.ac.uk.



Jinjun Chen is an Associate Professor from Faculty of Engineering and IT, University of Technology Sydney, Australia. He holds a PhD in Computer Science and Software Engineering (2007) from Swinburne, a master degree in engineering (1999) and a bachelor degree in applied mathematics (1996) from Xidian University, China. His research interests include cloud computing, social computing, green computing, service computing, e-science, workflow management.

He has published more than 100 papers in high quality journals and conferences, including TOSEM, TSE, and ICSE. He received IEEE Computer Society Outstanding Leadership Awards (2008-2009, 2010-2011), Vice Chancellor Research Award, and many other awards. Contact him at jinjun.chen@uts.edu.au.



Fei Wang is a master student in School of Computer Science and Technology, Huazhong University of Science and Technology. He received the B.S. degree from Wuchang Institute of Technology, Wuhan, China in 2010. He worked in Hubei Zhongyou Technology Industrial Co.Ltd. and Wuhan Tianyu Information Industry Co.Ltd. for two years after his graduation. His research interests include embedded system and big data analysis. Contact him at wgf110@gmail.com.



Fei Hao is an assistant professor in Huazhong University of Science and Technology. He received the B.S. and M.S. degrees in School of Mathematics and Computer Engineering from Xihua University, Chengdu, China, in 2005 and 2008, respectively. He was a research assistant at Korea Advanced Institute of Science and Technology and Hangul Engineering Research Center, Korea. He is studying toward the PH.D degree in the Department of Computing at the University of Bradford, Bradford, UK. He has

published over 30 research papers in international and national Journals as well as conferences. His research interests include social computing, big data analysis and processing and mobile cloud computing. Contact him at feehao@gmail.com.



Man Lin received the B.E. degree in Computer Science and Technology from Tsinghua University, China, 1994. She received the Lic. and Ph.D degrees from the Department of Computer Science and Information at Linkopings University, Sweden, in 1997 and 2000, respectively. She is currently an associate professor in Computer Science at St. Francis Xavier University, Canada. Her research interests include system design and analysis, power aware scheduling, optimization algorithms. Her research is supported

by NSERC (National Sciences and Engineering Research Council, Canada) and CFI (Canada Foundation for Innovation). Contact her at mlin@stfx.ca.



Changqing Luo received his B.E. and M.E. degree from Chongqing University of Posts and Telecommunications in 2004 and 2007, respectively, and the Ph.D. from Beijing University of Posts and Telecommunications in 2011, all in Electrical Engineering. During his Ph.D. study, he worked as a visiting student in Dept. of Electrical and Computer Engineering at University of British Columbia (UBC) for half year, and also in Dept. of Systems and Computer Engineering at Carleton University for half year. After the graduation, he joined the school of Computer Science and Technology, Huazhong University of Science and Technology in 2011, where he currently works as an Assistant Professor. His current research focuses on algorithms and optimization for wireless networks, cooperative communication, green communication, resource management in heterogeneous wireless networks, and mobile cloud computing. He has served on the Technical Program Committee (TPC) of several conferences, such as the TPC member of IEEE ISWTA'13, ICC'13-GMCN, ICC'12, PIMRC'12. Contact him at chqluo2013@gmail.com.



Laurence T. Yang received the B.E. degree in Computer Science and Technology from Tsinghua University, China and the PhD degree in Computer Science from University of Victoria, Canada. He is a professor in the School of Computer Science and Technology at Huazhong University of Science and Technology, China, and in the Department of Computer Science, St. Francis Xavier University, Canada. His research interests include parallel and distributed computing, embedded and ubiquitous/pervasive computing. His research has been supported by the National Sciences and Engineering Research Council, and the Canada Foundation for Innovation. Contact him at ltyang@gmail.com.